

**NASA Contractor Report 3915**

NASA-CR-3915 19850026280

# **Robot Environment Expert System**

**Jerry L. Potter**

**GRANT NAG1-341  
AUGUST 1985**

RECEIVED  
JUL 11 1985

U.S. GOVERNMENT  
PRINTING OFFICE

U.S. GOVERNMENT PRINTING OFFICE  
WASHINGTON, D.C. 20540  
1985-0026280

**NASA**



NASA Contractor Report 3915

# Robot Environment Expert System

Jerry L. Potter  
*Kent State University*  
*Kent, Ohio*

Prepared for  
Langley Research Center  
under Grant NAG1-341



National Aeronautics  
and Space Administration

Scientific and Technical  
Information Branch

1985



## 1. INTRODUCTION

The REES system which was developed under grant NAG-1-341 is an expert system designed to aid a ground based operator in the control of a space borne robot. It consists of a sensor input component, a database component and a query answering component integrated into a single cohesive system. As indicated in the original REES proposal, the envisioned system is much too complex to implement in a single year. Thus the intent of this initial effort was to prove the feasibility of the basic concepts.

The development of REES involved four phases. First, the overall system design was defined, establishing the specifications of the three system components. Second, the required hardware configuration was established. Third, the software was designed, coded, debugged and integrated with the hardware into a working system. Finally, a set of demonstration tasks were designed, coded, debugged, and demonstrated.

Naturally, in an undertaking with limited resources, certain short cuts had to be taken. Nevertheless, over 3000 lines of code were delivered implementing the full functional capability of the original proposed design and demonstrating that the REES design is a viable approach for monitoring a remotely controlled robot. However to be a complete, well functioning system, more development is needed. Also, it is important to note that REES can easily be augmented with new sensor types (such as range finders), object modeling capabilities and automatic image processing techniques as desired.

The basic concepts behind REES and its overall design are described in Appendix A.

## 2. REES DEMONSTRATION

The Demo program developed at Kent State and delivered to NASA demonstrates the essential elements of REES. This program tests the basic concepts and exercises the major systems of REES demonstrating how each aspect of REES can be implemented. In particular, it demonstrates:, 1) the ability to establish a viable REES configuration, 2) sensor movement through an unknown environment, 3) data base generation from TV imagery, 4) data base refinement, 5) data base querying for path planning, and 6) the automatic detection of motion by the expert system component.

### 2.1 The REES Configuration

A viable REES configuration consists of two sub-configurations: 1) a REES environment configuration capable of being monitored via a mobile sensor, and 2) a hardware configuration enabling the integration of video and graphics imagery at an operator console. The later configuration was especially important since it is the crux of a remote, manually operated, computer based robot environment expert system. That is, it demonstrates that remote teleoperated robot systems can be built and controlled using todays technology with no automatic image processing or scene analysis. However, the REES functional design allows such capabilities to be easily added as they become available.

The REES environment used for the demo programs was specified by positioning the origin at universal coordinate (1300, 300, 100) and the maximum coordinate values to (2100, 1100, 900). Thus the REES coordinates range from 0 to 800 in

each dimension (including time). See Figure 2.1-1.

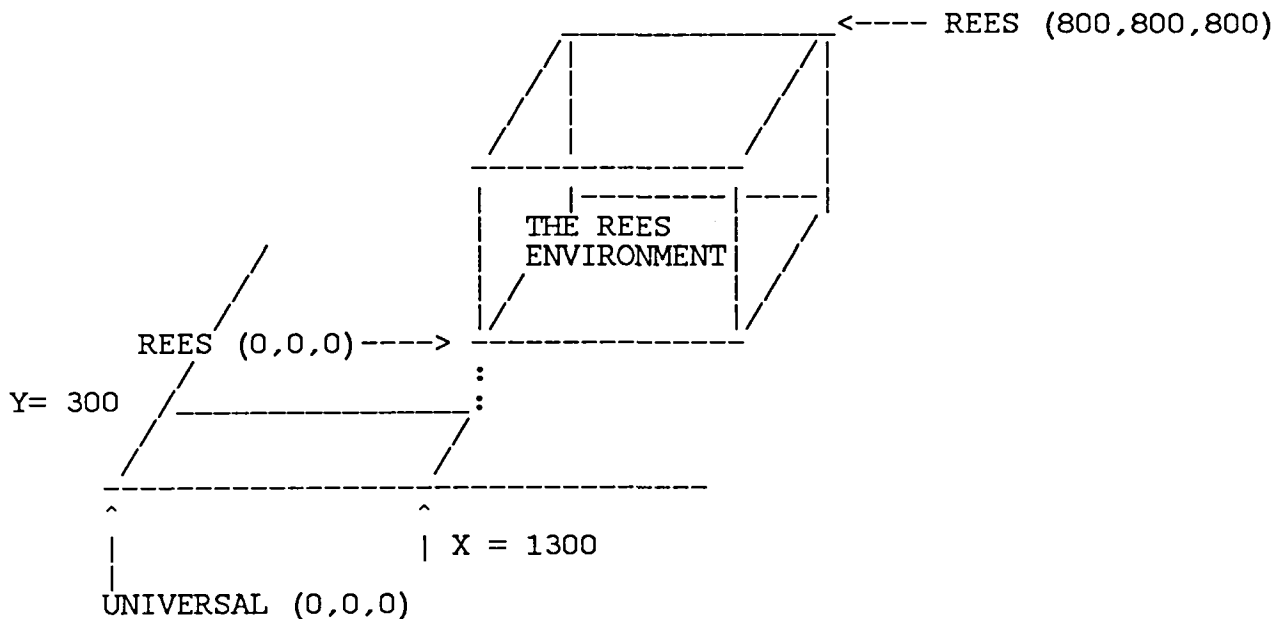


Figure 2.1-1 The Position of the REES Demo Environment

It was necessary to restrict the environment to the area shown in Figure 2.1-1 due to the limitations of the Unimate arm. The Unimate was designed to position a grasping end effector at a specified point at a specified orientation. However, the Unimate software accepts the two situations shown in Figure 2.1-2 as equivalent. Obviously, while these positions may be equivalent for grasping, they are not equivalent for imaging. Moreover, the Unimate's joint system was designed to allow the manipulation of objects in an envelope surrounding it (See Figure 2.1-3). The arm is designed so that within this envelope, the orientation of the end effector is intended to be pointing essentially outward. The Unimate is capable in theory of positioning the end effector in any direction, but the joint configuration severely restricts

the ability of the arm to orient the sensor to look inward as illustrated in Figure 2.1-4.

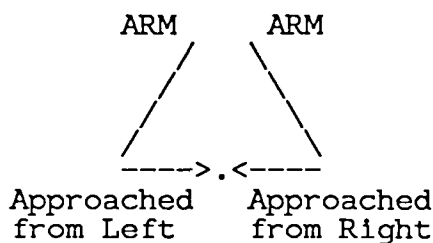


Figure 2.1-2 Equivalent End Effector Positioning

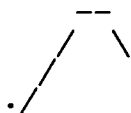


Figure 2.1-3  
Manipulator Envelope



Figure 2.1-4  
Inward Looking Sensor

In order to effectively establish a REES environment it is necessarily to be able to view all portions of it from multiple positions. This requirement is difficult to achieve given the above limitation of the Unimate arm. Consequently, the multiview capability was established by restricting the environment to the region illustrated in Figure 2.1-1 rather than the entire universal coordinate space as originally planned. It should be pointed out that this limited environment is due to the physical



limitation of the current robot system and in no way is due to the REES design. Certain modifications in REES such as a cylindrical rather than a rectangular coordinate system may improve the situation but a properly conceived and implemented arm for sensor positioning is the only permanent solution.

Figure 2.1-5 illustrates the hardware configuration of the REES system. Fundamental to the system is the operator's monitor display (VS11). The basic display is the TV imagery generated by the sensor. The REES system generates properly scaled graphical representations of regions (cells) in the REES environment and displays them over the TV imagery so that the operator can correctly identify and position objects in the environment.

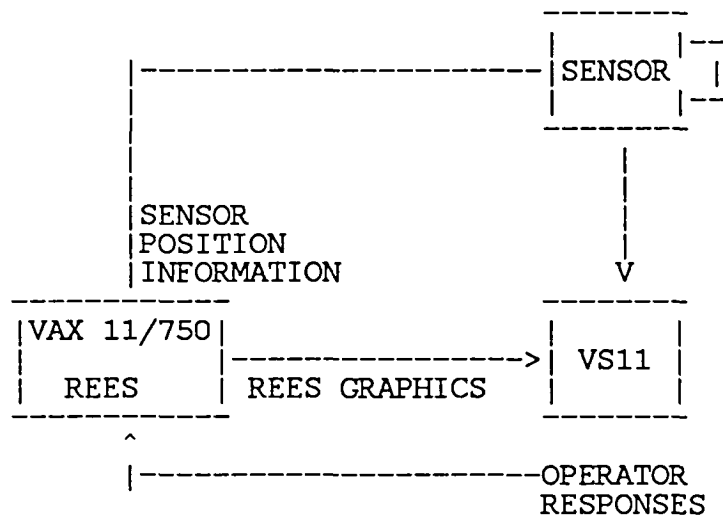


Figure 2.1-5 REES Hardware Configuration

## 2.2 Sensor Positioning in an Unknown Environment

When the demo program is started, the contents of the

environment are unknown. In order to avoid collisions with unknown objects, the REES system uses a bootstrapping approach for sensor positioning. The only assumption is that the initial camera/arm position is empty. From then on, all avenues for sensor movement (or "corridors") are verified by manual inspection before they are used. To demonstrate this capability, the REES demo first positions the sensor to a pre-specified location with the sensor oriented to look down a specific corridor. The outline of the cells composing the corridor are displayed graphically over the TV imagery of the environment. The operator determines if the corridor is empty or not by visual inspection and tells REES. If the corridor is not empty an alternative path is determined.

This step demonstrates the effective use of operator interaction in place of less accurate and more costly automatic scene analysis and object recognition. However, as technology in these areas improve, this step could be done "semi-automatically" (i.e. the computer automatically determines the results with operator override).

### 2.3 Data Base Generation

REES is based on the assumption that pre-prepared data bases are unreliable. Therefore, it was essential to demonstrate the ability to generate the data base directly from the environment. After a corridor has been verified as clear as described in

Section 2.2, the sensor can be safely positioned anywhere in the corridor.\* At each position in the corridor,\*\* the sensor imagery is analyzed interactively by the operator. REES uses the results of this editing process to generate the REES environment database. Figure 2.3-1 shows the position of the objects in the data base.

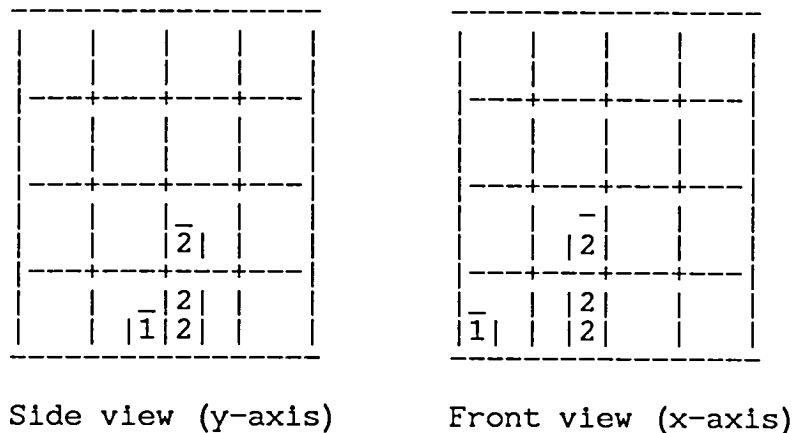


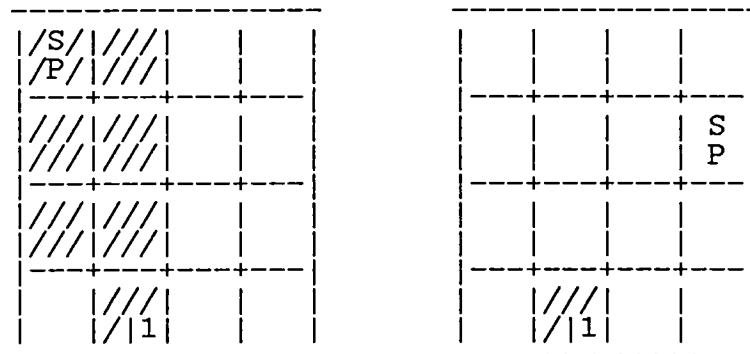
Figure 2.3-1 Box 1 and Box 2 Positioning

After the camera is in position and the scene is displayed on the VS11, the operator outlines, names and edits the objects of interest.\*\*\* After each object in the imagery is identified,

-----  
 \*Unfortunately, due to arm positioning limitations, the corridor displayed (and verified as clear by the operator) is not the one used subsequently for data base generation.

\*\*In order to minimize the work involved in processing the imagery in the demo environment only two sensor positions are used.

\*\*\*There are two regions records for each object. One for a left view, one for a right view. However, due to the small distance involved in the demo environment, the information from both views would be redundant so only one, the right, is used in the demo. In addition, due to the inability to get the VS11 WAITSWITCH routine to function correctly, the object outlining and editing step must be done before hand in a stand alone mode. The data file resulting from the editing process is read in at this time instead of actually being generated on line.



// indicates cells associated with box 1  
SP = sensor position

Initial Database

Refinement

Figure 2.3-2

Figure 2.4-1

REES projects them onto their REES cells and stores the cell ids and object associations in the data base. As can be seen from Figure 2.3-2 false cells may be associated with the objects. The next section, data base refinement illustrates how these extraneous cells are eliminated.

## 2.4 Data Base Refinement

In order to minimize storage and processing time, REES is designed to position and outline objects to the highest (largest) level allowable. For example, the results of Figure 2.3-2 (ie. Box 1 is in cells (6) (4 6) (4 2) (4 0)) is adequate if the true area of concern is cell (2) or (0). See appendix C for a cell map.

However, requirements always change so it is necessary to be able to refine the data base, restricting the objects to their correct REES cells. In the REES demo, the next step is to position the camera to cell (2 0) for database refinement. A second view is displayed on the VS11 from this position. After

the operator outlines, names and edits the objects, REES refines or restricts the REES cells to the correct subset of cells (See Figure 2.4-1). (in order for this step to function properly, the operator must label, i.e. identify, the objects consistently in the various views). These two views are able to define the demo data base sufficiently well to allow REES to avoid the objects during path planning.

## 2.5 Path Planning

Now that the objects, box 1 and box 2 have been defined in the REES demo data base, the path planning capability is demonstrated by positioning the camera in cell (0 3) in front of box 2 and commanding the sensor to be moved to cell (4 5) behind the box. If the camera moved directly, it would knock the box over. The path planning routine queries the data base to determine where obstacles are and plans a path around them, positioning the camera at cell (4 5) without knocking over box 2.

## 2.6 Motion Detection

During the data base generation and refinement phases, a velocity component was assigned to boxes 1 and 2. Since the boxes did not move during these phases, the velocity components entered were zero. During this last step of the demo, the camera is positioned at cell (6 6) to view box 1.

Box 1 is manually moved to a new position (cell (4 4)) representing the fact that it is starting to move (it no longer

has a velocity of zero). When the operator identifies box 1 in the editing phase, REES attempts to refine the cell definition of box 1 further. However, it is unable to do so since there are no overlapping cell projections due to the fact that box1 is in an entirely different cell. REES concludes that it has detected a discrepancy, deduces that the environment has changed, warns the operator, issues an emergency stop for all manipulators under its control and awaits further instructions.

### 3. RESULTS

The hextree data organization was used for modeling the robot environment during the demonstration process described in Section 2. This data organization allows REES to "understand" three dimensional space and time. The REES data base was implemented in LISP and was integrated with a data base generation and querying capability. Appendix B describes the details of the hextree data organization, its use and its advantages.

The path routine of the query portion of the REES system was developed to a sufficient degree to prove effectiveness. That is, it will find a path of a specified minimal clearance between any two points in the environment. This routine can be used as the basis for a robot activity planner. The basic path finding capability can be enhanced so that it can be instructed to find only certain classes of paths. For example, it may be desirable to restrict arm movement to an area "in front" of an object. The path routine, cell content and object description routines provide the basis of a querying system which can be augmented

with rule based control structures to produce a complete robot environment expert system which knows where all the objects in the environment are and knows how these objects move and interact.

The major emphasis of this research was to demonstrate that the REES database could be generated, modified and/or verified by an operator on the ground. This is crucial since it is unrealistic to depend on the integrity of a priori ground generated data. Techniques to accomplish data base generation were developed and this capability was demonstrated in the laboratory. An important aspect of this design is its ability to work effectively with low resolution TV imagery. This is accomplished by a refinement technique which uses multiple views of an object to successively refine the objects shape and position.

Also demonstrated was the ability of REES to detect object movement and take preventive action. In particular, the last portion of the demo involved unexpectedly moving an object in the environment. When REES determined that the object is not in the predicted position it issues a halt command stopping further arm movements to avoid possible collisions. The operator can then inspect the situation and restart the system when the discrepancy has been corrected.

In addition to the cell data base, an initial object data base capability was also included. This data base provides the ability to describe attributes and facts about objects. Provision is made for the addition of expert system rules to be added to use these facts so that REES can gain a knowledge of

each individual object. This information can be used to predict how the objects behave and how they are to be manipulated. For example, if a satellite had left and right handed threaded bolts, the thread direction of each bolt would be noted in the database. Moreover, data on how the end effector must manipulate each thread type would be included. If the operator attempted to put a nut on a bolt by turning it in the wrong direction, REES would issue a warning before the threads were damaged.

#### 4. CONCLUSIONS

The hextree approach works well for positioning objects in space and time with any desired degree of resolution as described in Appendix B. Moreover, it also explicitly identifies open spaces which are available for the manipulator to move through. Also the shape of an object is described to any desired degree of resolution since each object in the data base has associated with it the cells which contain it. Algorithms have been developed which allow cell information to be manipulated (translated and/or rotated) and to be used for object recognition (See bibliography of Appendix B).

Remote data base generation, verification and update was demonstrated to be feasible using conventional graphic techniques and a conventional display device (the VS11) which allows video and graphics to be displayed simultaneously. Moreover, the technique demonstrated does not require a digitizer and is feasible with low quality black and white imagery.

Finially, while the need for a mobile sensor is unquestionable, it was evident from this research effort that the



concept of mounting a camera on a manipulator arm needs considerable more study. First, it is apparent that the nature of the movements required for an end effector are substantially different from those which are desirable for a vision sensor and that special joint configurations or arm positioning capability may be desirable. Second, it is natural to view objects of interest from a distance of a few feet not a few inches as would be necessary if the camera were mounted on the same arm as the end effector. Third, the best angle for viewing is not the best angle for end effector manipulation. Finally, there is vital information to be gained from viewing the end effector and object to be manipulated simultaneously. This would be difficult to do effectively if the camera is mounted on the same arm as the end effector. In conclusion the utility of REES was shown in the demonstration program and further refinement of the concept is recommended, however, though analysis should be spent on the utility of a separate arm for sensor positioning.

## APPENDIX A

### ROBOT ENVIRONMENT EXPERT SYSTEM

#### 1. INTRODUCTION

The Repair Environment Expert System (REES) is knowledgeable about the current configuration of the three dimensional working space around and in between both the robot servicer and its object of service such as a satellite in need of repair or on orbit construction. REES would be used for:

1. Monitoring all appendage (i.e. arm) movements to predict and avoid collisions with other appendages or any portion of the robot or object satellite.
2. Monitoring arm movement to aid in task execution. It can be used as the sensory feedback portion of a servo system.
3. Plan generation. REES can calculate the path between any two points or objects, since it knows where occluded objects and invisible/artificial restrictions such as radiation hazards, laser hazards, etc. are.
4. Fact verification. REES's knowledge of 3d space surrounding the robot and its object allows it to verify or reject facts about the specific configuration of a satellite. If a satellite has been repaired a number of different times, its exact configuration may not be well

documented. REES would be able to verify which of several repair procedures was used and the exact resulting configuration. For example, when previously repaired, a two foot protrusion was installed instead of the original 1 1/2 foot one. REES will verify that the protrusion is indeed 2 feet.

5. Band width reduction. In a typical ground controlled remote orbital servicer scenario, only a small portion of the environment is changing at any one time. REES would eliminate the need to transmit the entire frame since it knows what objects are moving and where they are, only the portions of imagery which have changed would need to be transmitted. In addition, since REES can monitor the robot's movements as stated in items 1 and 2 above, lower resolution TV and slower frame rates can be used since precise visual feedback is not as critical.

Central to REES would be a partitioning of the 3d environment into levels of cells. For example, at the top level, the entire space would be divided into eight cubes as shown in Figure 1-1.

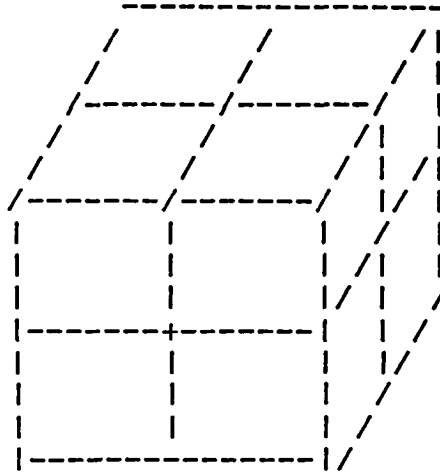


Figure 1-1 Top Level Cell Division

The objects and attributes of each cell would be noted. Those cells with important objects would be further divided into 8 subcells, again with contents noted. Those cells with unimportant or only one object would not be further subdivided. The exact contents of the data base would be a function of current need and would vary from application to application.

The first task of any application would be for REES to build its data base by exploring the environment with its sensors. Initially, only the top level cells would be defined and their contents unknown. Using interactive techniques so the operator can perform difficult pattern recognition and other judgmental functions, REES, using the robot's sensors, would fill in the details of each cell of interest.

After an environment has been defined, the data base can be saved and used as the initial data base in subsequent missions involving the same or similar environment. In these situations the data base would only have to be updated. The procedure would

be basically the same except that it would proceed faster since most operations would be simply verification of known facts.

One of the most useful sensors for defining the environment would be a TV camera mounted on an end effector. It has two advantages. First, the movement of the camera induces motion on the focal plane of the sensor which can be interpreted and used to define the cell's contents. Second, a hand mounted camera can get behind and/or take perspective views of objects to obtain additional positional information about the objects in a cell. These two advantages of a mobile camera allow a lower resolution sensor to be as effective as a higher resolution stationary camera. Moreover, the sequences of movement of the mobile sensor can be recorded so that the operator can later simply identify an object and REES will be able to remember how to move the sensor to view it.

A major advantage to having REES build and/or verify its own data base on station is that 1) it is extremely difficult to build such a data base a priori because the space between the satellite and robot will vary depending on docking location, docking orientation and minute details of the satellite itself, and 2) only the area of interest need be analyzed in detail instead of the entire environment space saving considerable memory requirements and cpu power to store and process unwanted data.

The expert system approach to dealing with the robotic environment has several additional advantages. It can accommodate in real time the changing environment of on orbit servicing and construction, including relative motion between the

servicer and its object. It can more easily accommodate or ignore intermittent errors and noise. It reduces the critical aspects of any transmission delay between the robot and the ground station and in addition it allows a reduction in bandwidth requirements.

This approach is also a first step in merging AI techniques with robotics and is easily extendible to take advantage of future advances in AI for automated operation, precision pointing and control, efficient data acquisition and real time data management where reductions in cost of 10 fold or greater are predicted ([14], p. 9).

Finially, probably the most important aspect of REES is that its basic concepts can be effectively proven with conventional robotic equipment using conventional simulation techniques. It is anticipated that the essence of the major components could be demonstrated working together in the laboratory with as little as 16 man months of effort expended over 12 months time. Individual major components could be demonstrated in as little as six months.

## 2. GENERAL

The Robot Environment Expert System (REES) is designed to be an expert on the three dimensional space in and around the robot and the satellite or construction being serviced. While in operation, the robot's environment will be constantly changing most obviously in a construction operation but just as certainly in a repair or service operation. Because of the temporal nature of the robot's environment, the REES must not only know where

every object currently is, but must know where they will be in the future.

The REES will be used by the human operator and other robot systems for such things as planning, tracking and collision avoidance. For example, during the planning phase for arm movement, REES will supply paths from point (object) a to point (object) b at time t. The planner can request paths for different times and between different objects and can then plan the optimal sequence of moves based on the information supplied by REES.

REES can also be used for tracking objects. Thus upon command, REES will cause the visual sensor (TV) to track the specified object (ie. an end effector) so that the object stays in the middle of the screen. Tracking is accomplished by locating the object of interest in each subsequent frame and then directing the visual system to look at the location of the object.

A similar concept but going forward in time is prediction. In this mode, REES will predict where an object will be at a given time based on the object's past history. This mode may be quite useful in complex construction tasks where the robot has to catch an object moving toward it.

Perhaps the most important use of REES will be collision avoidance. Whenever REES detects that an object is about to move into a cell occupied by another object, it will issue the necessary commands to avoid the collision. The normal way to perform collision avoidance of the robot arms or any other

movable peripheral under the robot's control would be to submit the peripheral commands to REES for approval just prior to sending them to the peripheral for execution. If REES detects any potential problems, it would prevent the commands from being sent.

Another method of collision avoidance is for REES to automatically internally track all moving objects via the robot's sensors. Whenever REES determines that two or more objects are about to occupy the same unit of space, it will cause avoidance action to take place.

### 3. REES Organization

The REES has three major parts, an expert component, a data base component and a perception component which are interconnected as shown in Figure 3-1. The expert component is the interface to the outside world, it will handle requests from the human operator and other robot systems. It not only must be able to interface with the outside world, but must also have sufficient knowledge of the data base component and expertise in real world dynamics to be able to answer all requests. It is the component that contains the knowledge that allows REES to predict the position of objects in the future. It knows the structure of the data base sufficiently well to be able to find paths from one object to another.



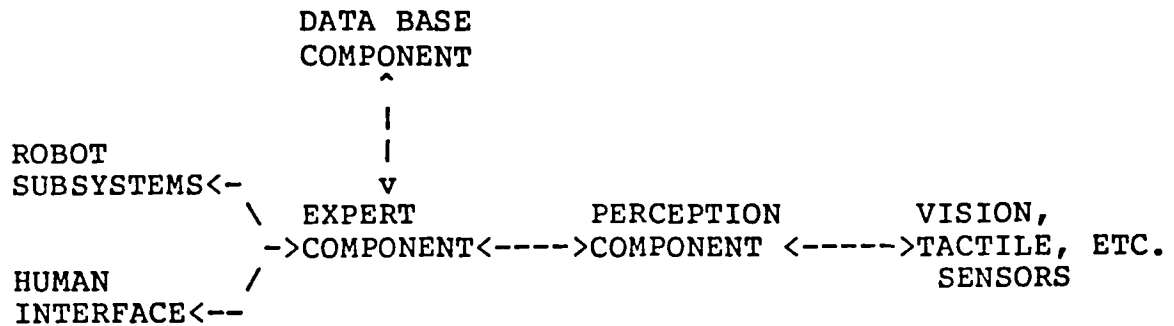


FIGURE 3-1 REES Component Interconnections

The data base component is basically a dynamic data base which not only keeps track of objects' positions but can redefine portions of its space and time metrics sufficiently fine to be able to accommodate objects of any size, moving at any speed. The most unique feature of the data base will be the variable quantization of the environment along both the spatial and temporal dimensions. Thus the space surrounding large objects will be quantized in large increments while the space around small objects will be quantized in small pieces. The motion of objects will also be monitored by a variable quantization of the space surrounding it. Thus the space in the path of a fast moving object will be quantized to a degree that allows its contents to be monitored every few seconds or milliseconds. Space in the path of a slow moving object will be quantized to allow monitoring at a slower rate. This variable quantization feature will save considerable memory space and most importantly cpu search time.

The perception component is a semi-autonomous subsystem. Its most important operation will be to automatically detect any changes of position via the robot's sensors. All changes

will be put into the data base. If a potential collision is detected during this operation, REES will notify the appropriate robot subsystem and the operator to take corrective action.

The expert system will be allowed to control the perception component for special functions. Most important perhaps is the interactive mode of operation where the operator and REES cooperatively identify objects and their position and put them in the data base. During this mode, it will be necessary to control the perception component in an intelligent manner so that all of the data present in the imagery and other sensors can be effectively utilized to build the data base. All three of these components are described in more detail below.

### 3.1 The Data Base Component

The data base component is the central repository of information for REES. It contains the data that the perception component extracts about the robot's environment. The data base is fed by the robot's sensors subsystems and therefore constitutes a true awareness of the robot's environment and is not simply a model or representation of it.

The data is organized into two separate structures, the coordinate data base and the object data base. The coordinate data base is a record of the attributes and contents of every subdivision of the coordinate space surrounding the robot servicer and its service object. The object data base is a record of the attributes of all objects within the robot environment.

### 3.1.1 The Coordinate Data Base

#### Variable Partitioning

The basic concept behind the Coordinate Data Base (CDB) is to simply divide the total volume of the robot environment into cells and then associate with each cell the attributes of the enclosed space. An important aspect of this approach is the ability to subdivide each cell recursively an arbitrary number of times so that any desired degree of precision can be obtained.

Figure 3.1.1-1 illustrates the optional subdivision capability. Cell subdivision is potent because only those portions of the robot environment of primary interest need be defined at the lowest level of detail. Thus the immediate work area may be subdivided to 5 or 6 levels so that the precision is of a scale suitable for hand manipulation. Surrounding cells may only be subdivided to 3 or 4 levels, sufficient for arm movement. While the outside cells may be subdivided only one level, sufficient for monitoring actions outside of the area of interest to such a degree that REES can notify other robot systems of dangerous and/or unusual situations with sufficient lead time to take corrective action.

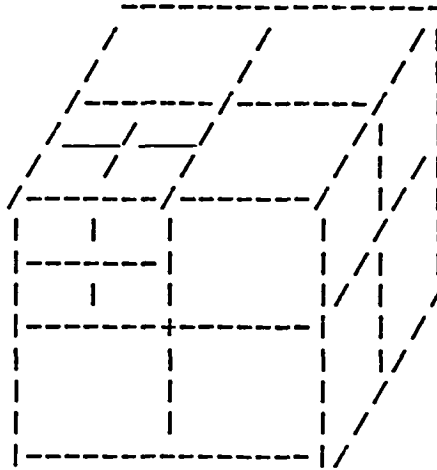


Figure 3.1.1-1 Optional Cell Subdivision

The variable levels of subdivision will save considerable amounts of storage and cpu time. If the entire robot environment had to be described to the smallest cell size, then a volume of 12'x6'x6' would have to be modeled to about a quarter of an inch or even smaller. This results in  $12 \times 6 \times 6 \times 12 \times 12 \times 12 \times 4$  or 2,985,984 subdivisions each one of which must be described with several words of memory and searched using several milliseconds of cpu time. With variable subdivision, the cubic foot surrounding the area of interest can be modeled to the .25 inch level with surrounding areas at larger intervals of 1 inch for the immediately surrounding 79 cubic feet and then 6 inch intervals for the remainder of the space (This approach is illustrated in Figure 3.1.1-2.). This results in only 144,832 items ( $12 \times 12 \times 12 \times 4 + 12 \times 12 \times 12 \times 79 + (12 \times 6 \times 6 - 80) \times 4$ ), over a 20 to one reduction of memory and processing capacity.

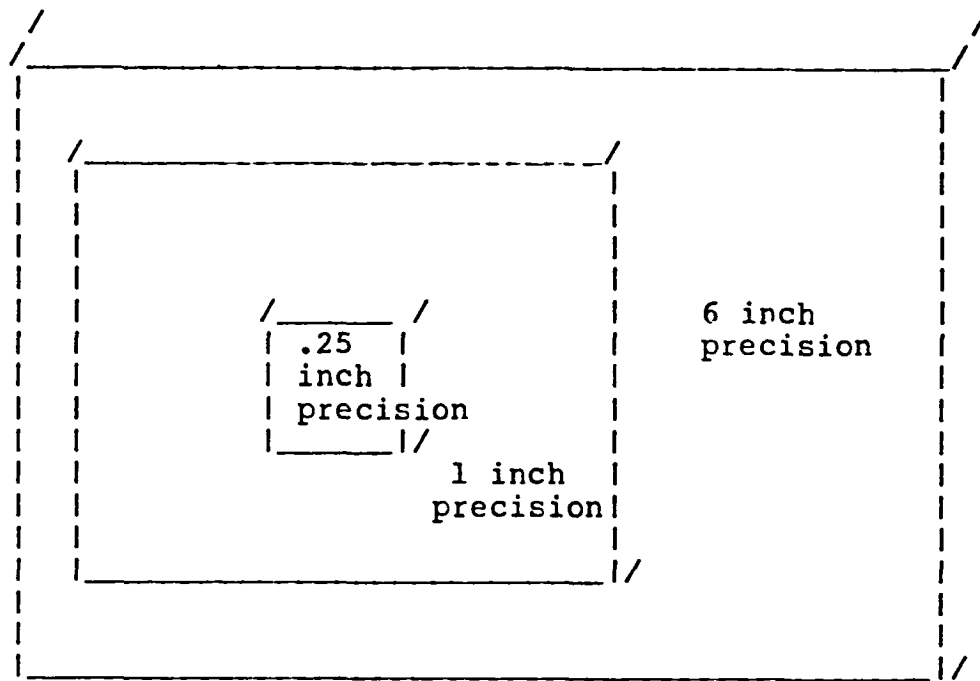


Figure 3.1.1-2 Variable Precision

#### Attributes

An attribute list is associated with each cell subdivision. These attributes pertain to the space inside the cell and characterize it with properties pertinent to the robot's sensors and effectors. For example, a cell may be characterized as solid, semi-solid or void (See Figure 3.1.1-3). If the cell is subdivided, the characterization of the subcells will be summarized at the higher cell levels. Thus if 10% of the subcells are solid and 90% are void, the cell may be characterized as 10% semi-solid, etc. However, caution must be exercised when doing this kind of analysis since by this approach, a solid metal cell would be labeled as only a few percent semi-solid if its subcells were subatomic in size.

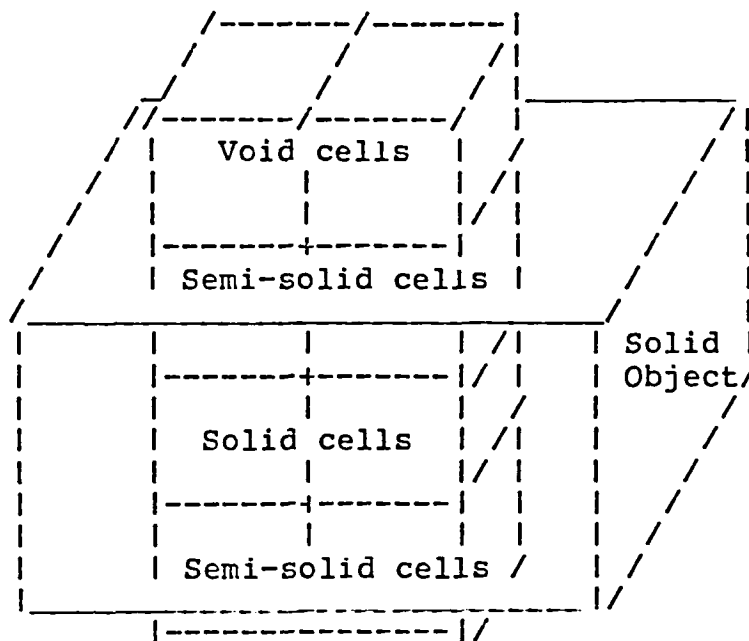


Figure 3.1.1-3 Cell Characterization

In addition to physical real objects which can be easily perceived by the robot's sensors, it will be important to recognize "invisible" obstacles such as radiation and laser hazards. If the robot has sensors for these "objects" then their extent can be precisely mapped in the same manner as ordinary objects. However, not all robots will have this capability. In some cases, the location of these hazards will have to be deduced from cues provided by other sensors. The attributes associated with the presence of these types of "objects" will be stored as a cell characterization attributes, i.e. radioactive, or laser hazard, in the same manner as solid objects are.

Attributes such as above which may be inferred and not directly perceived will be flagged so that the integrity of the data base can be ascertained if any conflicting data are detected.

Each cell will also have associated with it information as to how an object (i.e. arm or end effector) can move through the cell. This path information will consist of a description of adjacent subcells which are void (non solid, non radioactive and non laser hazard). The path data can be used by other robot subsystems and will be provided for all three directions of passage (front to back, top to bottom, side to side ).

An important attribute of each cell datum will be the description.validity time parameter. This attribute determines the time frame during which the datum description of the cell is accurate. Thus if the cell is in the path of a moving object, it may have several entries in the data base, one for each element of time leading up to , during and after the penetration of the cell by the object. Thus the cell x,y,z in Figure 3.1.1-4 would have three entries in the data base. One for time n-1, one for time n and one for time n+1. In actual practice, the number of entries in the data base for a cell would be a function of the time quantization requirements of the situation. Old time data, ie. cell entries with time entries older than current time, would be regularly deleted and new entries for cells that are to change in the near future would be regularly added in order to keep the data base up to date.

Finally, each cell will contain a list of the objects wholly or partially inside it.

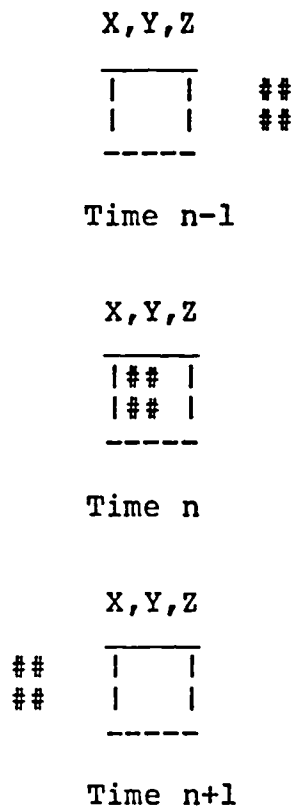


Figure 3.1.1-4 Time Varying Cell

#### Data Structure

The variable subdivision aspect of the CDB is very difficult to accommodate using the customary three dimensional array approach. In this approach, x, y and z coordinate values would be stored implicitly by the datum's position in the array as illustrated in Figure 3.1.1-5.



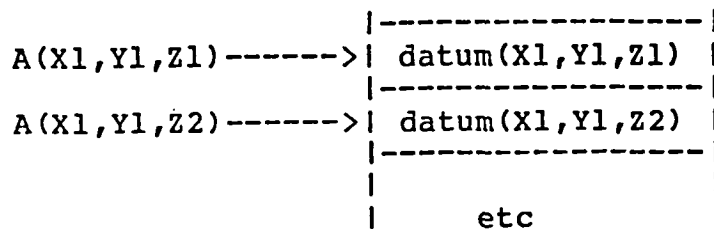


Figure 3.1.1-5 Implicit Addressing

Typically address algorithms for such storage arrangements such as:

$$A(x_i, y_i, z_i) = (x_i - 1) * \text{size}(y) * \text{size}(z) + (y_i - 1) * \text{size}(z) + z_i$$

require that memory locations for all possible address combinations of x, y and z be set aside. If this is done, however, the reduced storage and cpu advantages of the variable approach partitioning would be lost.

These advantages can be maintained by combining the Artificial Intelligence concept of data association with explicit datum addressing. Figure 3.1.1-6 shows a typical associative record with explicit addressing. By using this high level data storage technique, in addition to reduced storage and cpu processing time, considerable programming time can be saved. The associative record is easily modified to add, delete or change new fields. In addition, since all of the data is present explicitly, garbage collection is greatly reduced and in some instances can be eliminated altogether. Explicit address data records do not need to be sorted. These storage techniques are especially well suited for advanced hardware such as vector processors and associative memories.

X	Y	Z	CELL	CELL		CELL
ADDRESS	ADDRESS	ADDRESS	ID	ATTRIBUTE1		ATTRIBUTE N
-----	-----	-----	-----	-----	...	-----
-----	-----	-----	-----	-----		-----

Figure 3.1.1-6 Explicit Address Associative Record

### 3.1.2 Object Data Base

The Object Data Base (ODB) is the internal representation of the robot's world. In addition to a simple collection of objects, this data base contains information about the physical make up of the objects and the physical configuration of the robot's environment. This data base contains information about relative position of objects, object type, object properties and object size.

Associated with every object will be the ID of the object(s) in front of, behind, left of and right of it. In addition, the distance to these objects and the precision of the measurements will also be stored. Thus questions by the operator or by another robot system about the local environment of an object are easily answered. If more details are needed, the "in front of" and other properties can be chained forward by the expert component as far as necessary. The precision of the distances will be a function of the level of subdivisions of the cells involved.

The object type is an attempt to briefly describe the topology of the object. The "object" may be just a surface of a physical object, or it may be the actual object itself. If the object is essentially one dimensional, i.e., a pipe or wire, etc.

or essentially two dimensional, i.e. sheet metal, plate glass, etc, it will be so typed as one or two dimensional.

The object properties contained in the ODB are basically those needed for effective deployment of the perception component. However, the data will be available to whoever requests it. Typical properties are movability, pliability, transparency, color, color variations, etc.

Finially, the basic dimensions of the object are stored with an estimate..of the dimension's accuracy. Like the distance measurements described above, the object size measurement's accuracy will be a function of the cell subdivisions involved.

While the inexactness of the size and distance measurements may seem to be a hindrance to effective operation, it is actually an advantage since excessive detail and overly precise data is not generated or stored. If more precision is required, REES can easily obtain the data needed. It is a basic tenant of REES to store interally only that data which is essential for its immediate operation and only gather additional information and details as they are needed.

### 3.2 Perception Component

The limited resources aboard a space borne robot means that the perception component must be capable of operation with a minimum of computer and sensor equipment. The perception component (PC) of REES will be able to operate without automatic pattern recognition or high resolution sensors by using operator interaction and a moving sensor. The PC will position the sensor and request that the operator identify the object(s) of

interest. The PC will then induce motion on the focal plane of the sensor by moving it. This motion information will be analyzed and used by the PC to characterize the objects of interest. If the operator feels more information is necessary, he can direct the sensor to view the object from an entirely different perspective. The operator will identify in the new view the items which correspond to the objects identified previously. This interactive moving sensor approach to visual processing is an effective way to obtain the maximum amount of data in a low resolution sensor, low cpu power environment.

### 3.2.1 Operator Interaction

Key to the operation of REES is the philosophy of integration of tasks which are best performed manually with those which are best performed automatically. The most difficult tasks in the PC are 1) object recognition-scene analysis and 2) determining which objects are important to the task at hand and should be included in the data base. These tasks will be relegated to the human operator.

If the operator is to identify and organize objects there must be a method for allowing the operator to communicate with the PC. REES will use raster editing techniques for this purpose. The input from the visual sensor may have from 32 to 256 levels of gray and will be quite noisy. The first step in object specification is for the operator to interactively threshold the image until the object of interest is above the threshold value while the background is below. This can easily be done on a monitor with a light pen. The operator simply points to an area

inside the object, the system reads the grey value at that point and uses it as the initial threshold. The operator can then adjust the threshold with a trackball or joystick until the object is correctly delineated.

Next the operator must identify the pieces of the object since multiple areas may have been defined by the thresholding process. The operator again simply points to the regions of interest, the system automatically extracts the contiguous points in the area and overlays them on top of the original scene for verification by the operator.

The operator may add or delete areas from the object as needed until the overlay matches the object. He then identifies the object and signals that he is done. The system then associates in its data base, the label with the area designated.

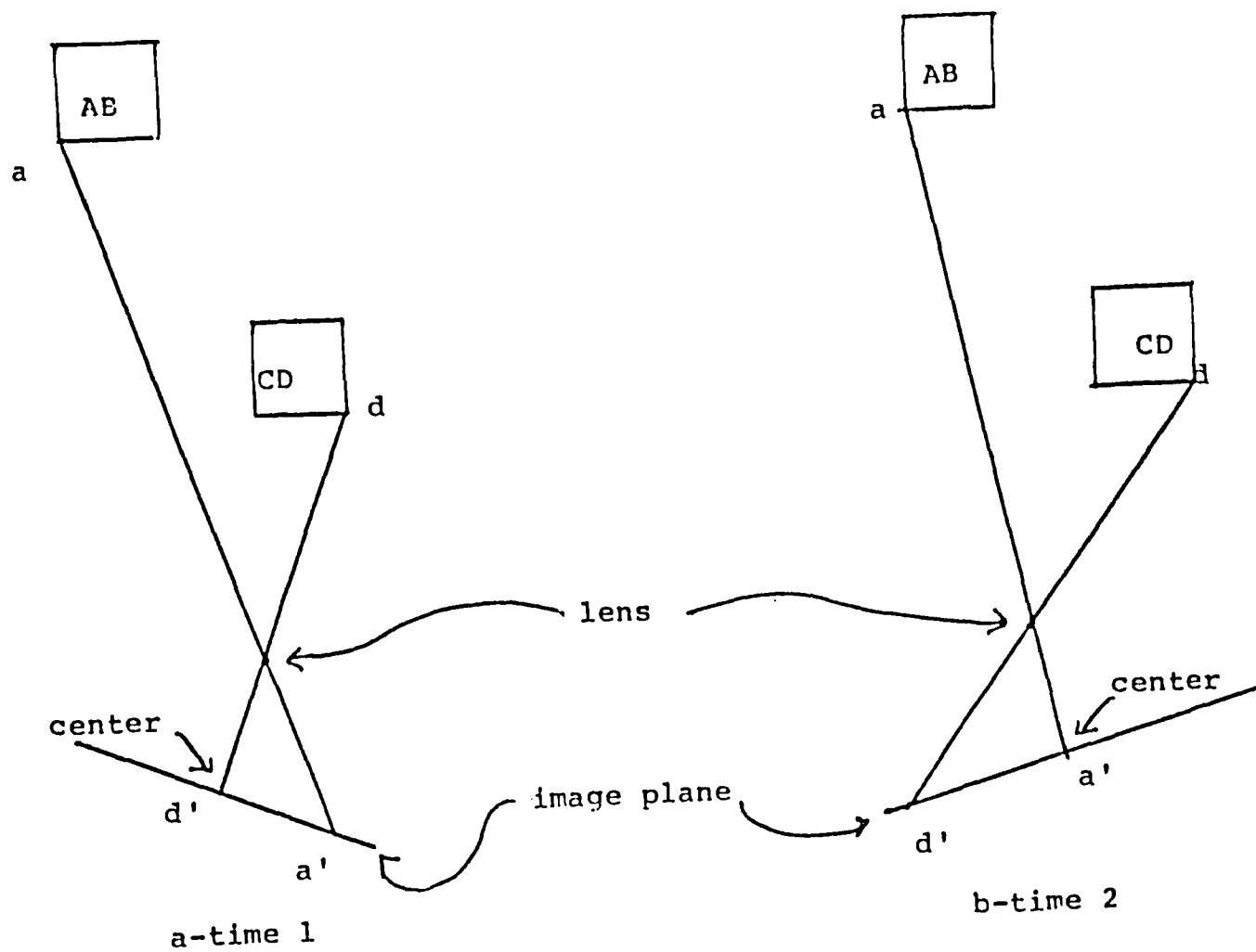
By using interactive process, the operator performs two important functions which are extremely difficult for REES to do, he identifies which objects are important and he delineates (or recognizes) the objects from the background in the scene.

### 3.2.2 Moving Sensor

The limited resources of the robot satellite will probably mean that only low resolution imagery can be used in space. This restriction can be helped by using a mobile camera. First of all, a mobile camera can move in as close to the object as required to obtain any degree of resolution. Second, the ability to obtain multiple perspective views of the same object should provide sufficient information about the objects position and shape to allow effective robot operation.

Loosely speaking, there are four types of sensor motion: none, jitter, head movement and locomotion. Jitter is the movement of the sensor lens through small angles about the sensor system (See Figure 3.2.2-1). Head movement is the movement of the entire sensor system about an external point of reference. Locomotion is the movement of the entire sensor system through great distances with no one specific point of reference. All three types of motion can be effectively used to extract information about a scene.

Figure 3.2.2-1 Effects of Jitter



## Motion Extraction

If motion is to be used to characterize objects, it must be more primitive than objects. One approach to motion processing is to identify the object first and then look for it to move. This approach has not been very successful because it requires powerful pattern recognition techniques to initially identify the objects. Algorithms sufficiently powerful to effectively recognize objects in stationary scenes have not yet been devised.

The alternative approach is to identify motion as a primitive or atomic feature independent of object recognition. One approach is to use template matching as described in [19]. Briefly, a template encompassing the basic shape of an unrecognized object is defined in the first frame of a sequence and then used to find a match in the next frame. The search starts at the point of definition in the first frame ( $x_i, y_i$ ) and proceeds concentricly outward until a match is found. The point at which the match occurs ( $x_i+n, y_i+m$ ) is subtracted from the defining point to produce a measurement of motion ( $n, m$ ) for the point and its associated but as yet unrecognized surrounding object.

This technique is quite robust and can be used for an arbitrary number of moving objects with occlusion and transforming shapes. The only requirement is that the frames be taken sufficiently close in time that the changes between frames are small (i.e. approximately continuous). Discontinuous changes such as an object on the left in the first frame and on the right in the second are time consuming to process and may lead to



erroneous results. This requirement is reasonable since it holds in the time band most important to humans (and therefore robots) and can be accommodated with conventional sensors.

Other approaches frequently require special purpose equipment and are based on a false model of the world. In particular they rely on continuous gray scale patterns in an image. But all images have discontinuities at the edges of objects which represent the most important information content of the scene, consequently these approaches are of little use. The template matching approach [19] can be used in all real world scene situations.

### 3.2.3 The Perception Process

The perception component must be capable of developing its data base from nothing and expanding its contents to the level needed at any particular point of time. This section will give a brief description of how this is done. This description will assume a mobile sensor attached to an end effector, however, the principles involved can be proven with a less mobile sensor.

The robotic environment (RE) will be partitioned into an initial set of cell subdivisions, the contents of which will all be labeled "unknown." The first step will be to point the sensor parallel to the surface of the RE and determine that the space (called the sensor space) in front of the RE is empty. The positioning of the RE and the sensor space will be done under operator control to avoid the possibility of the sensor colliding with any object (See Figure 3.2.3-1).

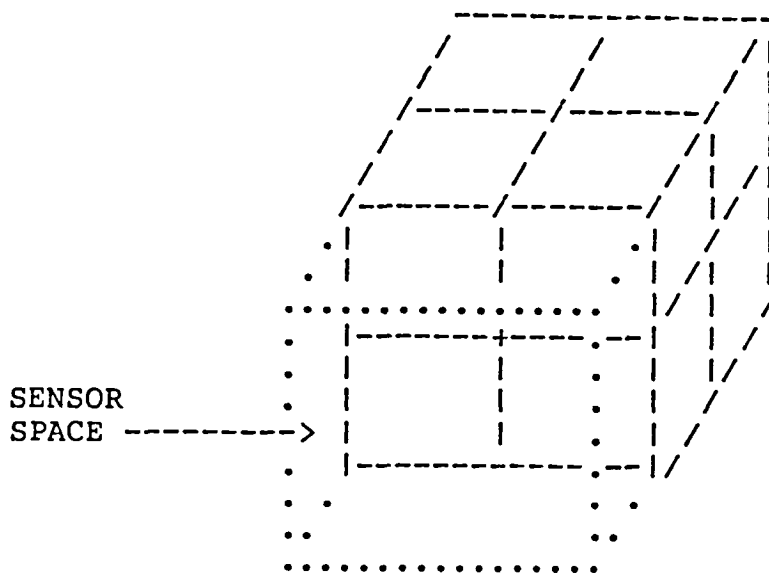


Figure 3.2.3-1 Sensor Space

Once the sensor space has been verified as clear, the sensor will move thru it to in front of the first subcell. The image of the first subcell will be analyzed interactively to the level specified by the operator, then the sensor will move on to the next subcell, where the process will be repeated. After the first tier of subcells have been processed, the second tier will be processed with the sensor space being the first tier of cells which have been sufficiently well characterized to allow the PC to direct the sensor thru it avoiding any objects. This process continues in an outside-in manner until all cells are occupied with solid objects. The process is then repeated from the beginning on a new surface of the RE until all surfaces have been processed.

### 3.3 The Expert Component

The Expert Component (EC) of the REES is the intelligent portion and is responsible for controlling the other components and answering questions from outside of REES about the robot's environment. It can perform tracking, position prediction and collision avoidance. It does not do planning, but would be used by a planner since when queried, it will respond with information about the position of an object, the distance between two objects, the objects at a specified location, the path between any two objects, and the relative position of any two objects. The robotic environment does of course vary with time. Objects move and are moved. Consequently, all of these functions and queries will be performed in a temporal mode.

The expert system will upon request track a specified object. This function can be used by the operator or other robot system to follow a specified object such as an end effector. If the operator requests that the vision system track the end effector, the REES will keep it in the center of the vision system avoiding the need for the operator to manually track it.

Tracking is following an object trying to keep up with it in real time. More sophisticated operations will require the knowledge of where an object will be in the future. For robotics systems which require this information, the EC will predict the position of an object at any given time in the future. The EC also supplies a parameter which specifies the degree of certainty of the prediction. The degree of certainty depends on the precision of the data base and the length of time in the future the prediction is to be made.

The REES will have built into it a collision avoidance capability. Inherent in the Cell Data Base is the ability to detect when an object is about to enter a cell which is already occupied by another object. The EC will monitor the CDB for such situations and will notify the appropriate robot subsystem of an impending collision. The time and method of notification will depend on the urgency of the situation.

The EC does not do planning, but does provide many useful functions for planners. It will calculate the distance between any two objects or points at any specified time. It will calculate a path in terms of subcell IDs from one object to another at a specified time. It will provide relative positional information (behind, above, etc) about objects at a given time. It will determine what objects are at a specified location either at a given time and/or will create a list of the objects which will occupy a position over some time interval. Finally, it will supply upon request all of the information it has about any object or cell in the data base.

The Perception Component is the primary input subsystem to REES, however, the EC also provides vital input information. Robot planner subsystems must notify REES via the EC of planned moves. This information is more accurate about the future than the perception system can provide and is crucial information for other planners if conflicts are to be avoided. The EC will decompose the plan and supply the information to the appropriate data bases.

In conjunction with this activity, each peripheral executive

should supply the EC with the next command to be sent. The EC will verify that the Robotic Environment is still receptive for such a move. If the RE has changed in a pertinent manner, during the planning and execution phases, the EC will prevent the command from being executed and will notify the planner of the need to modify its plans. If the move is acceptable, the EC will prime the perception component and data bases to expect the move. This priming of the PC and data bases will reduce computing power needs considerably.

#### 4. BIBLIOGRAPHY

- [1] Aggarwal, J. K., L. S. Davis and W. N. Martin, "Correspondence Processes in Dynamic Scene Analysis", PROCEEDINGS OF THE IEEE, vol. 69, no. 5, pp. 562-572, May 1981.
- [2] Barrow, H. G. and J. M. Tenenbaum, "Computational Vision", PROCEEDINGS OF THE IEEE, vol. 69, no. 5, May 1981, pp. 572-595.
- [3] Binford, T. O., "Survey of Model-Based Image Analysis Systems," THE INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH, Vol 1, No. 1, Spring 1982, pp.18-63.
- [4] Brice C. and C. Fennema, "Scene analysis using regions," ARTIFICIAL INTELLIGENCE, vol 1, no 3, pp. 205-226, 1970.
- [5] Clowes, M. B., "On seeing things," ARTIFICIAL INTELLIGENCE, vol. 2, no. 1, pp. 79-112, 1971.
- [6] Falk, G., "Interpretation of imperfect line data as a three-dimensional scene," ARTIFICIAL INTELLIGENCE, vol. 4, no. 2, pp.101-144, 1972.
- [7] Gilbert, A. L., "Video Data Conversion and Real - Time Tracking," COMPUTER, August 1981, pp. 50-56.
- [8] Granrath, D. J., "The Role of Human Visual Models in Image Processing," PROCEEDINGS OF THE IEEE, vol. 69, No. 5, May 1981, pp. 552-561.
- [9] Guzman, A., "Computer recognition of three-dimensional objects in a visual scene," MAC-TR-59 (thesis), Project MAC, M.I.T., Cambridge, MA, 1968.
- [10] Hill, J. and W. T. Park, "Real Time Control of a Robot with a Mobile Camera," PROCEEDINGS OF THE 9th INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ROBOTS, March 13-15, 1979, Washington D.C., pp. 233-246.
- [11] Horn, B. K. and B. G. Schunck, "Determining Optical Flow," ARTIFICIAL INTELLIGENCE, 17, 1981, pp.185-203.
- [12] Jacobus, C. J., R. T. Chien and J. M. Selander, "Motion Detection and Analysis of Matching Graphs of Intermediate - Level Primitives," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. PAMI-2, no. 6, Nov. 1980, pp. 495-510.
- [13] Jain, R., "Dynamic Scene Analysis Using Pixel Based Processes," COMPUTER, August 1981, pp. 12-18.
- [14] "Machine Intelligence and Robotics: Report of the NASA Study

Group. Final Report", March 1980, NASA CR-163380.

- [15] Marr, D and T. Poggio, "Cooperative computation of stereo disparity," SCIENCE, vol. 194, pp. 283-287, 1971.
- [16] Nitzan, D., A. E. Brain and R. O. Duda, "The measurement and use of registered reflectance and range data in scene analysis," PROC. IEEE, vol. 65, pp. 206-220, 1977.
- [17] Potter, J. L., "Extraction and Utilization of Motion in Scene Description," Ph.D. Thesis, University of Wisconsin, 1974.
- [18] Potter, J. L., "Velocity as a Cue to Segmentation," IEEE TRANSACTIONS SMC, May, 1975.
- [19] Potter, J. L., "Scene Segmentation by Velocity Measurements Obtained with a Cross-Shaped Template," PROCEEDINGS IV INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1975, pp. 803-810.
- [20] Potter, J. L., "The STARAN Architecture and its Application to Image Processing and Pattern Recognition Algorithms," AFIPS CONFERENCE PROCEEDINGS, 1978 NATIONAL COMPUTER CONFERENCE, June 1978, pp. 1041-1047.
- [21] Potter, J. L., "Continuous Image Processing on the MPP," PROCEEDINGS OF THE 1981 IEEE COMPUTER SOCIETY WORKSHOP ON COMPUTER ARCHITECTURE FOR PATTERN ANALYSIS AND IMAGE DATABASE MANAGEMENT, Hot Springs, Virginia, Nov. 11-13, 1981.
- [22] Potter, J. L., "Scene Segmentation Using Motion Information," COMPUTER GRAPHICS AND IMAGE PROCESSING, 6, 1977, pp. 558-581.
- [23] Roberts, L. G. "Machine perception of three-dimensional solids," in OPTICAL AND ELECTRO-OPTICAL INFORMATION PROCESSING, J. T. Tippett ET AL., eds. Cambridge, MA:M.I.T. Press, 1965.
- [24] Rosenfeld, A., R. A. Hummel and S. W. Zucker, "Scene labeling by relaxation operations," IEEE Trans. Syst. Man Cyberm., vol SMC-6, pp. 420-433, 1976.
- [25] Rubin, S. M., "Natural Scene Recognition Using Locus Search," COMPUTER GRAPHICS AND IMAGE PROCESSING, 13, 1980, pp. 298-333.
- [26] Thompson, W. B., "Combining Motion and Contrast for Segmentation," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 2, no. 6, pp. 543-549, 1980.

APPENDIX B

USING HEXTREES TO MODEL 4 SPACE

by

J. L. Potter  
Mathematical Sciences Department  
Kent State University  
Kent, Ohio 44242

ROBOTICS



## USING HEXTREES TO MODEL 4 SPACE

### Abstract

The Robot Environment Expert System uses a hexadecimal tree data structure to model a complex robot environment where not only the robot arm moves, but the robot itself and other objects may move. The hextree model allows dynamic updating, collision avoidance and path planning in time avoiding moving objects.

### 1. INTRODUCTION

Over the past several years, techniques have been developed for generating, manipulating and characterizing objects in the quadtree and octree data organizations. Algorithms have been described for generating quadtree representations from boundary data and binary arrays [Samet 1980a and 1980b], and generating octrees representations from two dimensional slices [Yau 1983a]. Manipulation algorithms such as rotation, translation and set operations have been described [Hunter 1979, Jackins and Tanimoto 1980]. Techniques for characterizing objects described in the octree format have been developed [Samet 1981 and Schneir 1981]. The 2 and 3 dimensional aspects of the quadtree and octree model has been extended to 4 dimensions ( a hexadecimal or hextree) to model time varying images[Udupa 1982]. Due to this wide spread interest the quadtree and octree representations have been generalized to n dimensions [Yau 1983b].

Another important but less studied application of the octree

This research is supported in part by NASA Grant NAG-1-341.

and related data structures is to model 3d space not 3d objects. That is, the working environment of a robot can be modeled by placing objects in the various cells of an octree data structure which models the encompassing 3d space. This slight but important variation on object modeling is crucial in robotic applications where it is necessary not only to model object size and shape but equally important object position and the voids between objects so that arm movements can be planned. A modified octree approach has been described which increases storage efficiency and allows planning of robotic arm movement [Lozano-Perez 1981]. This approach uses cells of varying size and position so that the cell boundaries may be more accurately aligned with object boundaries resulting in reduced storage requirements and greater positional accuracy.

The hextree model used by REES generalizes the octree concept to 4 dimensions so that objects moving in the environment can be modeled. This enables REES to predict (and thus avoid) collisions of all objects in the environment including arms and manipulators under its control and external moving objects over which it has no direct control. Since the entire environment in space and time is modeled, paths for manipulator movement can be planned which avoid both stationary and moving objects.

## 2. BACKGROUND

The Robot Environment Expert System (REES) is a space borne system which facilitates the efforts of a remotely controlled satellite servicing robot. It is designed to be an expert on the space in and around the robot and the satellite or construction

being serviced. REES is intended to monitor movements for collision avoidance, monitor movement for positional feedback, help in planning robot movements and aid in database generation and verification. Consequently, REES must not only know where every object currently is, but must know where they will be in the future.

### 3. HEXIDECIMAL TREES

#### 3.1 Representation and Labeling

The data base is organized into two separate structures. The cell data base and the object data base. The object data base is a record of the attributes of the objects in the robot environment and will not be described further here. The cell data base is a record of the attributes and contents of every cell subdivision of the space surrounding the robot servicer and its service object.

The basic concept behind the cell data base is to simply divided the total volume of the robot environment into a hierarchy of cells and then associate with each cell the attributes of the enclosed space. An important aspect of this approach is the ability to subdivide any individual cell an arbitrary number of times so that any desired degree of precision can be obtained. Figure 3.1-1 illustrates the optional subdivision capability (for 3 dimensions).

In the hextree data organization, the time dimension and the space dimensions are treated the same. Similarly to three dimensions, where the top level of the octree represents the entire region of concern (environment) and the next level

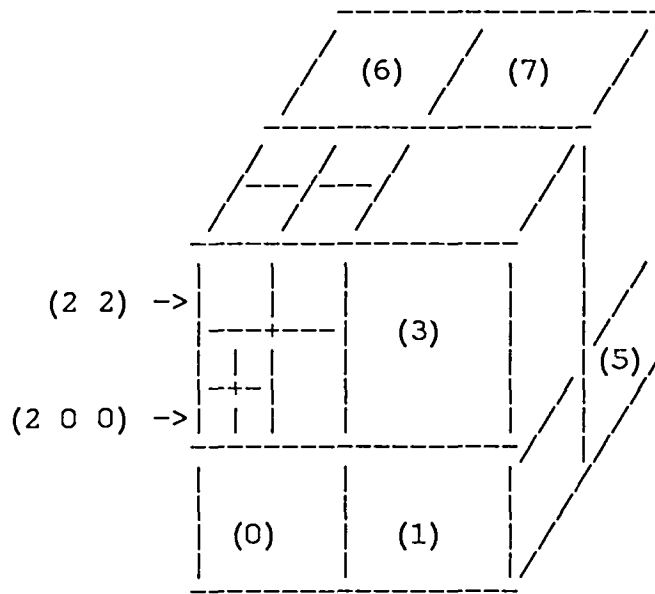


Figure 3.1-1 Variable Subdivision

represents the subspaces obtained by dividing all dimensions in half, in four dimensions, the top level cell includes not only all space but all time and the second level contains subcells which are not only  $1/8$  the size ( $1/2$  size in each dimension) of the original cell but contain only  $1/2$  of the time interval. Thus sixteen cells are needed to represent the entire subdivision, 8 cells for the first half of the time interval and 8 cells for the second half.

The label of a cell consists of a list of numbers which specifies the hierarchy of cells from the root to the specified cell. The label of the entire space, i.e. the top level cell, is the nil list  $()$ . The label of the lower, left, front subcell at level one is  $(0)$ . See Figure 3.1-1. The label of the lower, left, front subcell of cell  $(0)$  is  $(0\ 0)$ , a level 2 cell. In general, the label is a list  $(d_1\ d_2\ d_3\ \dots\ d_i)$  where  $i$  is the

level index and  $d_i$  is the subcell code for that level. In the hextree organization, the 16 cells are identified by a hexadecimal digit from 0 to F. The digits from 0 to 7 represent the cells during the first half of the divided time interval and the digits from 8 to F represent the same 3 space cells during the second half. Thus cells (0) and (8) represent the same physical location but at different time intervals.

In order to allow easier illustrations, the time representation will be described using 2d space. Hexadecimal digits 0 to 3 will be used for the first time interval and 8 to B will be used for the second time interval. Given the 2d cellular space shown in Figure 3.1-2a at time  $t_1$  and the same space shown in Figure 3.1-2b at time  $t_2$ , the corresponding tree structure is shown in Figure 3.1-3a and b. The cells of Figure 3.1-2 are labeled corresponding to the positions of  $T_1$  and  $T_2$  shown on the bottom time line of Figure 3.1-3a. Figure 3.1-3b shows the tree structure that corresponds to the time subdivisions. Figure 3.1-5 shows the octree associated with Figures 3.1-2 and 3.1-3.

### 3.2 Paths, Objects and Moving Objects

The hexadecimal tree organization deals effectively with stationary and moving objects and paths. As shown in the previous section, a stationary object is inside several cells of the hextree. That is, since time as well as space is divided at each level of the tree, a stationary object in cell (0) during the first half interval, it will be in cell (8) during the second half. If the tree is further divided to say 3 levels, then a stationary object which is in cell (0 0 0) is also in cells (0 0 8), (0 8 0), (0 8 8), (8 0 0), (8 0 8), (8 8 0) and (8 8 8).

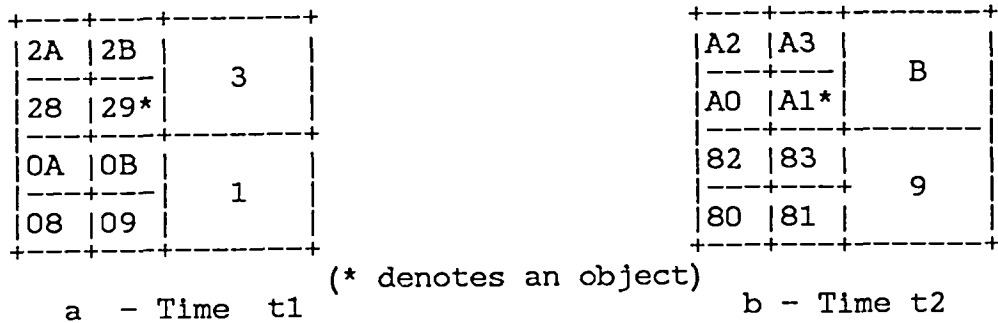


Figure 3.1-2 - Two space at two different times

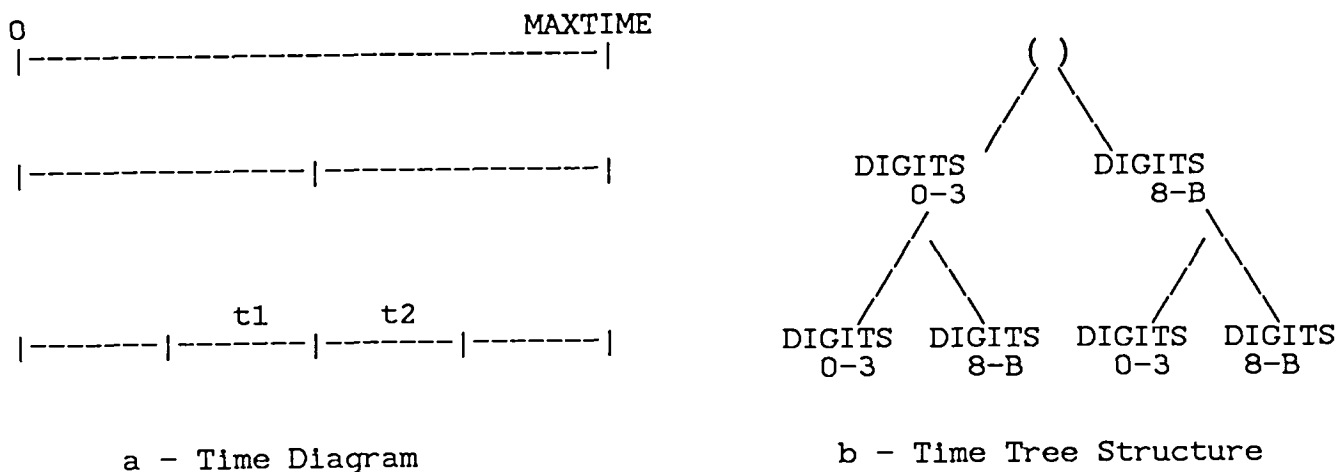


Figure 3.1-3 Time Representation

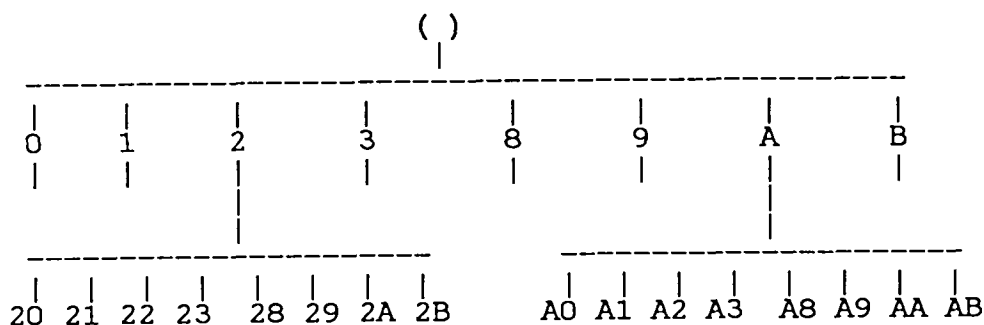


Figure 3.1-4 - A Partial Octree for Two Dimensions plus Time

Since the cells are hierarchical, the object is also in cells (0 0), (0 8), (8 0), (8 8), (0), (8) and ().

Given the diagram of 3 space shown in Figure 3.1-1, if a moving object is in cell (0) during the first half of the time subdivision and in cell (1) during the second half, then the object is in cells (0) and (9) in hextree notation. Similarly if an object is moving so that it is in the 3 space cells (0 0 0), (0 0 1), (0 1 0), (0 1 1), (1 0 0), (1 0 1), (1 1 0) and (1 1 1) at time increments 0, 1, 2, 3, 4, 5, 6 and 7 respectively, then in hextree notation it is in cells (0 0 0), (0 0 9), (0 9 0), (0 9 9), (9 0 0), (9 0 9), (9 9 0) and (9 9 9). Note that in hextree notation both stationary and moving objects are in a sequence of cells. That is, there is no distinction between moving and non-moving objects except for the cells they occupy.

Similarly, a planned path is the same as a moving object. That is, it is represented as a sequence of cell ids. The list ((1 1) (1 9) (9 1) (9 9)) is a path in time represented in 3 space octree notation by the stationary cell (1 1). When a path is planned, it reserves a cell for the specified "time slot." Even though paths, objects and moving objects all are represented by sequences of cell ids, paths are treated differently because they are under system control while object position and movement are not. That is, a path is a "planned or future" object that does not exist outside of the current time cell since the system can always change its mind. An object on the other hand is real and will occupy the future time cells unless specific action is taken to stop it. Thus the distinction between paths and objects is only for administrative convenience not because of any difference in their physical representation in the database.

ADDOBJECT accepts a cellid and an object specification. If the object is entirely or partially inside the cell in question and the cell has subcells, ADDOBJECT is called recursively on the subcells and the density of the current cell is calculated as a function of the density of the subcells. If the cell is a leaf its density is set to one and the density value is returned. If the object is entirely outside of the cell, the existing cell density is returned.

```

ADDOBJECT (CELLID,OBJECT)

IF OBJECT IS INSIDE CELL THEN
  ADD OBJECT TO OBJECT LIST
  IF SUBCELLS EXIST THEN
    DENSITY = 0
    FOR I = 1 TO F
      DENSITY = DENSITY + ADDOBJECT(CELLID+"I",OBJECT)
    ENDFOR
    DENSITY = DENSITY/8
  ELSE DENSITY = 1
  IF DENSITY GT 1 THEN COLLISION
  RETURN DENSITY
ELSE RETURN EXISTING DENSITY OF CELL

```

Figure 3.3-1 - PDL for ADDOBJECT

### 3.3 DENSITY

A measurement of the degree to which a cell is occupied, its "density," is associated with each cell. Whenever a path or object is entered into the data base (See Figure 3.3-1), it is added in a recursive top down manner until a leaf cell of the hextree (the lowest level cell) is reached. The lowest level cell is set to full (density = 1). The densities of the ancestor cells are calculated as a percentage of the lower level cells which are full. Thus a middle level cell may contain several different objects and paths and still not be full (i.e. its density is less than one).

The density parameter is used to detect collision situations. As long as the density of a cell is less than or equal to one there is no conflict or potential collision.



Consequently during the object or path addition process, the density of each cell is monitored. Whenever the addition of an object to the data base causes the density of a cell to exceed one, a warning is issued specifying the object(s) and cell(s) involved.

### 3.4 PATH

One of the basic REES functions is to find a path consisting of a set of empty cells from a specified cell to a second specified cell. Note that the PATH algorithm is not a planner. It does not determine the best path given a criteria and perhaps more importantly for robotic applications, it returns a path in four space for a single unjointed object - i.e. a wrench, etc. It does not return a path for a complex jointed object - i.e. a robot arm.

Given a graph, a common algorithm for finding a path between any two nodes is to systematically search the neighboring nodes of the two starting nodes, generating two expanding circles of nodes until the two circles intersect (i.e. have one or more common nodes). If a record is kept of which node lead to which node, a path from one of the common nodes to both original nodes can be obtained by back tracking.

This algorithm can be applied to the hextree path problem since the cells and subcells can be represented as nodes of a graph and the adjacent surfaces of cells and subcells can be represented as edges. However, since cells of the hextree can be subdivided, the graph representation must be applied recursively as shown in Figure 3.4-1.

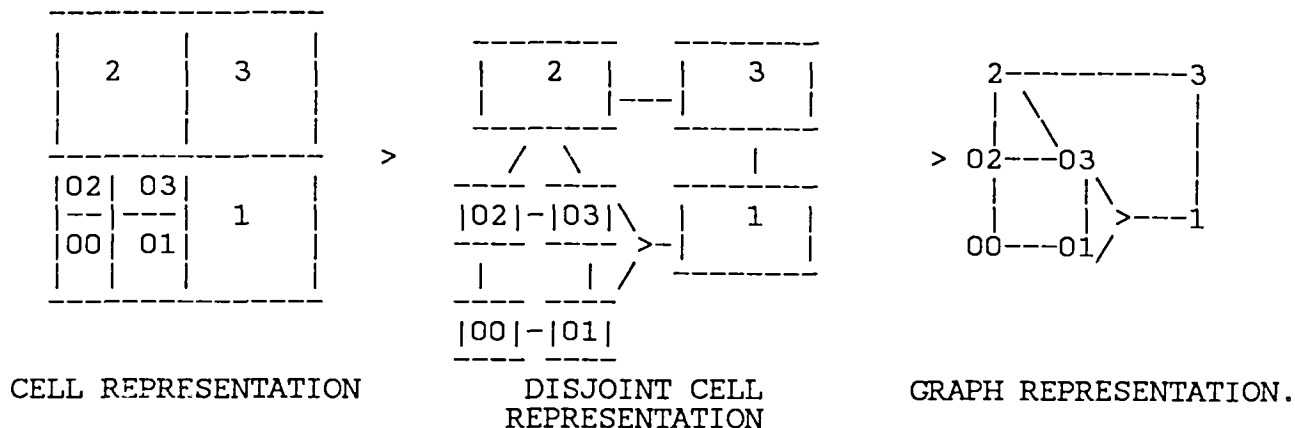


Figure 3.4-1 Recursive Graph

The data base is dynamic and is not in graph form. That is, the edges are not explicitly identified and the mapping between the nodes and cells may change. For example, at one time cell (0) may be a node while at other times it may be replaced by the cells (0 0), (0 1), (0 2) ... (0 F). The digits in the neighboring cell's id are calculated from the input cell's id proceeding from the least significant to the most significant by the following formula:

$$\text{ODIGIT} = \text{MODULO}_{2^* \text{INC}} (\text{IDIGIT} + 3 * \text{INC} + \text{CARRY})$$

The increment value, INC, is a function of the direction as shown in Table 1. The carry into the least significant digit is zero but succeeding carry's are calculated by either of the two formulas below as indicated in Table 1. [ ] indicate truncation.

$$(1) \quad \text{CARRY} = [\text{ODIGIT} / (2^{**}(\text{INC}-1))] * \text{INC}$$

$$(2) \quad \text{CARRY} = (1 - [\text{ODIGIT} / (2^{**}(\text{INC}-1))]) * \text{INC}$$

Carry out of the most significant digit indicates that the boundary of the environment has been reached. That is, an "out of range" cell id has been calculated.

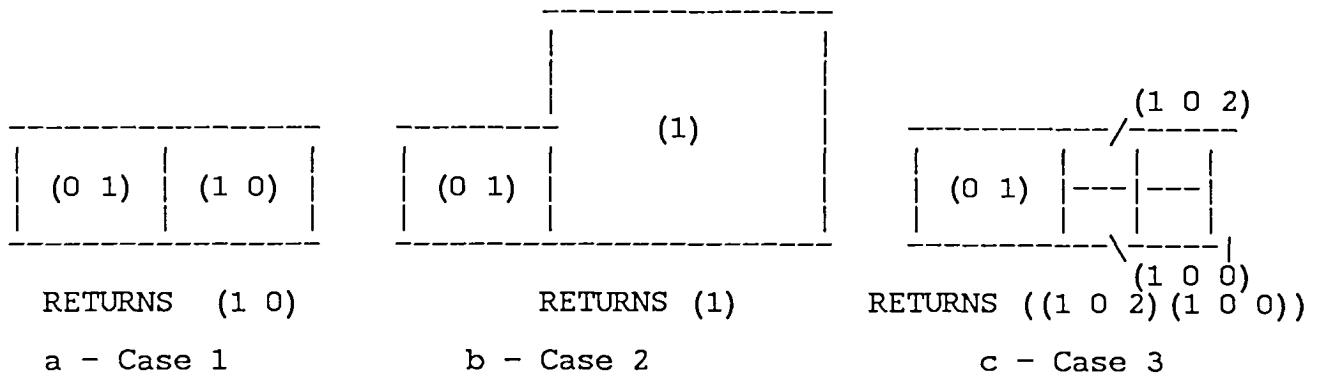


Figure 3.4-2 Neighboring Cells for Cell (0 1)

The path search algorithm operates by expanding both nodes looking for a common cell. When an intersection is found, the path is obtained by backtracking to the nodes. In order to prevent backtracking in negative time, the Path routine expands all nodes which descend from the origination node in forward time only. The nodes descended from the destination node are expanded in backward time only. Thus when a path is found it will move from the origination to the destination cell in positive time increments.

#### 4. EXPERIENCE

The hextree data structure for REES has been implemented in FORTRAN and LISP on a VAX 11/750 at the Intelligence Systems Research Laboratory of the Automation Technology Branch of the NASA Langley Research Center. Due to the large computing requirements of the image analysis portion of the system, REES could only simulate real time detection of moving objects. However, all reactions and path planning were correct and faithfully executed in the simulated time frame.

TABLE 1

DIRECTION	INCREMENT	CARRY FORMULATION
EAST (RIGHT)	1	(1)
WEST (LEFT)	1	(2)
NORTH (UP)	2	(1)
SOUTH (DOWN)	2	(2)
BACKWARD	4	(1)
FRONTWARD	4	(2)
PLUSTIME	8	(1)
MINUSTIME	8	(2)

The calculated neighboring cell id may not exist in the data base, may not be empty, or may be the empty sibling of a larger empty cell. All of these cases are checked for and a list of the largest empty cells greater than the specified minimum size which are "related" to the input cell and are the first cells to be reached from the direction specified is returned. That is, the returned list consists of 1) the cell itself, 2) an ancestor cell or 3) a list of descendent cells on the side closest to the neighboring cell. Figure 3.4-2 illustrates these three cases.

The uniformity of the data representation allows the path routine to search in time as well as space. For example, assume that a path is sought from cell (0 0) to cell (9 8). Note that the cells are separated by one unit of space and two units of time. If a moving object is in cell (0 9). That is, if the intervening space cell is filled during the first time interval, then the path routine will find the path consisting of waiting one time unit and then moving to the desired cell (i.e., the path returned would be (0 0) (0 8) (8 1) (9 8) ).

The hextree data structure used by REES allowed it to model both stationary and moving objects in the environment. The regular size of the cell subdivisions allowed objects and paths to be modeled in space and time. The hextree approach is economical since it lets a volume to be modeled by a large single cell during the time interval that it contains no object and by smaller higher resolution cells during the time interval that it does contain objects. Moreover, the hextree data base was easily modified to accommodate unforeseen changes in the environment as they occurred. Since "free space cells" were calculated at run time directly from the data base, valid collision free paths were generated. In conclusion, the hextree data organization is a promising approach to model time varying environments for robotic applications.

## 5. REFERENCES

- Hunter, G. M. and K. Steiglitz, "Operations on Images Using Quad Trees," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, PAMI-1, Vol. 2, April 1979, pp. 145-153.
- Jackins, C. L. and S. L. Tanimoto, "Oct-trees and Their Use in Representing Three-dimensional Objects," COMPUTER GRAPHICS AND IMAGE PROCESSING, Vol. 14, 1980, pp. 249-270.
- Lozano-perez, T., "Automatic Planning of Manipulator Transfer Movements," IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, Vol. SMC-11, No. 10, October 1981, pp. 681-698.
- Samet, H., "Region Representation: Quadrees from Boundary Codes," COMMUNICATIONS OF THE ACM, Vol. 23, no. 3, March 1980, pp. 163-170.
- Samet, H., "Region Representation: Quadtree from Binary Arrays," COMPUTER GRAPHICS AND IMAGE PROCESSING, Vol. 13, 1980, pp. 88-93.
- Samet, H., "Computing Perimeters of Regions in Images Represented by Quadrees," IEEE TRANSACTIONS OF PATTERN ANALYSIS AND MACHINE INTELLIGENCE, PAMI-3, No. 6, November 1981, pp. 683-687.

- Schneir, M., "Calculations of Geometric Properties Using Quadtrees," COMPUTER GRAPHICS AND IMAGE PROCESSING, Vol. 16, 1981, pp. 296-302.
- Udupa, J. K., S. N. Srihari and G. T. Herman, "Boundary Detection in multidimensions," IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, PAMI-4, 1982, pp. 41-50.
- Yau, M. J., HIERARCHICAL REPRESENTATION OF THREE-DIMENSIONAL DIGITAL OBJECTS, Technical Report No. 201, SUNY Buffalo, April 1983.
- Yau, M. J., and S. N. Srihari, "A Hierarchical Data Structure for Multidimensional Digital Images," COMMUNICATIONS OF THE ACM, Vol. 26, No. 7, July 1983, pp. 504-515.









1. Report No. NASA CR-3915		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Robot Environment Expert System				5. Report Date August 1985	
				6. Performing Organization Code	
7. Author(s) Jerry L. Potter				8. Performing Organization Report No.	
9. Performing Organization Name and Address Kent State University Mathematical Sciences Department Kent, OH 44242				10. Work Unit No.	
				11. Contract or Grant No. NAG-1-341	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				13. Type of Report and Period Covered Contractor Report Jan. 1983 - May 1984	
				14. Sponsoring Agency Code 506-54-63-01	
15. Supplementary Notes  Langley Technical Monitor: Nancy E. Orlando Final Report					
16. Abstract  The Robot Environment Expert System uses a hexadecimal tree data structure to model a complex robot environment where not only the robot arm moves, but also the robot itself and other objects may move. The hextree model allows dynamic updating, collision avoidance and path planning over time, to avoid moving objects.					
17. Key Words (Suggested by Author(s))  Robot environment, Expert system, Hextree, Machine vision			18. Distribution Statement  Unclassified - Unlimited  Subject Category 62		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 60	
				22. Price A04	



National Aeronautics and  
Space Administration

Washington, D.C.  
20546

Official Business

Penalty for Private Use, \$300

**BULK RATE**  
**POSTAGE & FEES PAID**  
NASA Washington, DC  
Permit No. G-27



**POSTMASTER:** If Undeliverable (Section 158  
Postal Manual) Do Not Return

---